

Penelitian

**PERANCANGAN PENYEIMBANG ROBOT
OTOMATIS BERBASIS MIKROKONTROLLER
ATMEGA16A**



MULKAN ISKANDAR NASUTION, M. Si
NIDN. 2004047601

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2017**

KATA PENGANTAR

Bismillahirrahmanirrahim...

Dengan segala kerendahan hati, penulis sampaikan puji syukur kepada Allah SWT berkat Rahmat dan Hidayah-Nya memberi kesehatan, pengetahuan dan kesempatan kepada penulis sehingga dapat menyelesaikan penelitian ini yang berjudul **“Perancangan Penyeimbang Robot Otomatis (*Self Balancing Robot*) Berbasis Mikrokontroller Atmega16a”**.

Dalam menyelesaikan penelitian ini banyak bantuan bimbingan dari berbagai pihak, baik berupa materil, spiritual, maupun informasi. Sehingga penelitian ini dapat diselesaikan. Maka selayaknya penulis mengucapkan terima kasih kepada :

1. Bapak Dr. H.M. Jamil, MA selaku Dekan Fakultas Sains dan Teknologi UIN Sumatera Utara Medan
2. Ibu Dr. Rina Filia Sari, M.Si selaku Wakil Dekan I Fakultas Sains dan Teknologi UIN Sumatera Utara Medan
3. Bapak Dr. Abdul Halim Daulay, ST, M. Si selaku Kaprodi Fisika Fakultas Sains dan Teknologi UIN Sumatera Utara Medan sekaligus Konsultan pada penelitian ini.

4. Bapak/ibu rekan-rekan dosen tetap Fakultas Sains dan Teknologi UIN Sumatera Utara Medan.

Menyadari kekurangan dan keterbatasan pada penelitian ini, maka penulis tetap mengharapkan kritik dan saran dari berbagai pihak agar penelitian ini bisa dikembangkan dikemudian hari.

Akhir kata semoga penelitian ini bermanfaat bagi kita semua dan semoga Allah SWT berkenan memberikan berkahNya sehingga semua harapan dan cita-cita penulis dapat terkabulkan. Amin.

Medan, Oktober 2017

Medan, Oktober 2017
Konsultan

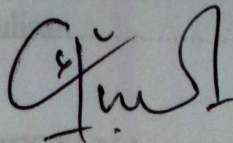
Mulkan Iskandar Nasution, M. Si

Dr. Abdul Halim Daulay, ST, M.Si
NIP.19811106 2005 01 1003

REKOMENDASI

Setelah membaca dan menelaah hasil penelitian yang berjudul **“Perancangan Penyeimbang Robot Otomatis (*Self Balancing Robot*) Berbasis Mikrokontroller Atmega16a”**. Yang dilakukan oleh Mulkan Iskandar Nasution, M. Si maka saya berkesimpulan bahwa hasil penelitian ini dapat diterima sebagai karya tulis berupa hasil penelitian. Demikianlah rekomendasi diberikan kepada yang bersangkutan untuk dapat dipergunakan sebagaimana mestinya.

Medan, Oktober 2017
Konsultan



Dr. Abdul Halim Daulay, ST, M. Si
NIP.19811106 2005 01 1003

2.3.1 Kontrol Proporsional.....	12
2.3.2 Kontrol Integral.....	13
KATA PENGANTAR.....	i
REKOMENDASI.....	ii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	vi
DAFTAR TABEL.....	vii
ABSTRAK.....	viii

BAB 1. PENDAHULUAN

1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah	2
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	3
1.6. Sistematika Penulisan	3

BAB 2. TINJAUAN PUSTAKA

2.1. Penyeimbang Robot Otomatis	6
2.2. Complementary Filter	7
2.2.1 Integration	9
2.2.2 Low Pass Filter	9
2.2.3 High Pass Filter.....	10
2.2.4 Sampling Periode.....	10
2.2.5 Waktu Konstan.....	10
2.3. Pengendali PID	11

2.3.1 Kontrol Proporsional	12
2.3.2 Kontrol Integral.....	13
2.3.3 Kontrol Derivatif.....	13
2.4. Mikrokontroller ATmega	16
2.5. Sensor MPU6050	18

BAB 3. PERANCANGAN ALAT DAN PROGRAM

3.1. Diagram Blok.....	21
3.2. Perancangan Rangkaian Mikrokontroller ATmega16	23
3.3. Rangkaian MPU 6050	25
3.4. Rangkaian Regulator.....	30
3.5. Rangkaian H-Bridge	30
3.6. Flow Chart Program.....	39

BAB 4. PENGUJIAN ALAT DAN PROGRAM

4.1. Pengujian Mikrokontroller ATmega16	41
4.2. Pengujian Gyroscope	42
4.3. Pengujian Accelerometer	47
4.4. Pengujian Complementary Filter	51
4.5. Pengujian PID	55

BAB 5. KESIMPULAN DAN SARAN

5.1. Kesimpulan	57
5.2. Saran.....	57
Daftar Pustaka.....	59

DAFTAR GAMBAR

1. Gambar 2.1. Pendulum terbalik di atas kereta beroda ..	6
2. Gambar 2.2. Self balancing robot beroda dua menyeimbangkan diri.....	7
3. Gambar 2.3. Blok diagram complementary filter	8
4. Gambar 2.4. Block diagram PID kontrol	14
5. Gambar 2.5. Pin konfigurasi	18
6. Gambar 2.6. MPU6050	19
7. Gambar 3.1. Diagram block rangkaian	21
8. Gambar 3.2. Rangkaian mikrokontroller ATmega16 ...	24
9. Gambar 3.3. Axis sudut accelerometer	25
10. Gambar 3.4. Axis kecepatan sudut.....	28
11. Gambar 3.5. Rangkaian modul GY-521	29
12. Gambar 3.6. Rangkaian Regulator.....	30
13. Gambar 3.7. Rangkaian H-Bridge.....	32
14. Gambar 3.8. Flowchart program	39
15. Gambar 4.1. Hasil Pembacaan gyroscope.....	46
16. Gambar 4.2. Hasil Pembacaan Accelerometer.....	51
17. Gambar 4.3. Hasil pembacaan complementary filter....	53
18. Gambar 4.4. Hasil pembacaan Accelerometer, Gyroscope dan complementary filter.....	54
19. Gambar 4.5. PWM saat robot dalam keadaan stabil berdasarkan perhitungan PID	56

DAFTAR TABEL

Tabel 3.1. Tabel Kebenaran Gerbang NAND.....	33
Tabel 3.2. Tabel Kebenaran IC IR 2111	33
Tabel 3.3. Tabel kebenaran Rangkaian H. Bridge	38
Tabel 4.1. Hasil Pembacaan <i>Gyroscope</i>	46
Tabel 4.2. Hasil Pembacaan <i>Accelerometer</i>	50
Tabel 4.3. Hasil Pembacaan Complementary Filter	53

ABSTRAK

Penyeimbang robot otomatis (*self balancing robot*) adalah simulasi transportasi beroda dua di bagian kanan dan kiri robot yang seterusnya disebut ROBA-G1, ROBA-G1 didesign untuk mempertahankan posisinya pada posisi seimbang pada permukaan datar dengan cara bergerak maju dan mundur. Agar ROBA-G1 dapat mempertahankan posisi seimbang maka pada pembuatan robot ini digunakan sensor *accelerometer* dan *gyroscope* yang kemudian output kedua sensor tersebut digabungkan dengan *complementary filter* untuk mendapatkan sudut yang seharusnya. Pergerakan motor yang halus sangat dibutuhkan pada pembuatan *self balancing robot* maka diperlukan sistem kontrol PD untuk memperhalus pergerakan robot dan menekan drift. Hasil yang didapat robot dapat mempertahankan posisi pada posisi seimbang dengan toleransi $\pm 3^0$. ROBA-G1 dapat mencapai posisi seimbang dari kemiringan $\pm 10^0$.

Kata kunci : *Penyeimbang robot otomatis, sensor accelerometer, gyroscope, complementary filter, toleransi,*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penyeimbang Robot Otomatis (*Self balancing robot*) beroda dua merupakan robot yang memiliki dua buah roda di sisi kanan dan kirinya yang tidak akan seimbang apabila tanpa kontroler. *Self balancing robot* yang diberi nama ROBA-G1 ini merupakan pengembangan dari model pendulum terbalik (*inverted pendulum*) yang diletakkan di atas kereta beroda. Menyeimbangkan robot beroda dua memerlukan suatu metode kontrol yang baik dan handal untuk mempertahankan posisi robot dalam keadaan tegak lurus terhadap permukaan bumi tanpa memerlukan pengendali lain dari luar. Penelitian dan pengembangan self balancing robot meningkat pada akhir-akhir ini. Sejak Dean Kamen, 2001 telah mendeklarasikan SEGWAY sebagai "*The first Selfbalancing, electric powered transportation device*" Kemudian diikuti oleh penelitian penelitian sejenis di berbagai negara. Hal ini tentunya dipicu oleh perkembangan teknologi elektronika, robotika serta IT yang cukup pesat di seluruh dunia.¹

Untuk mendesain dan membangun ROBA-G1 beroda dua yang mampu menyeimbangkan dirinya yang tegak lurus terhadap permukaan bumi di daerah bidang datar. Pada

¹Bambang Nurcahyono, Sumardi, Budi Setiyono, Self-Balancings Scooter Menggunakan Metode Kendali Proporsional Integral Derivatif, 2010

Penelitian ini digunakan mikrokontroler AVR ATMEGA16, sensor *accelerometer* dan *gyroscope* serta control Proporsional Integral dan Derivatif (PD) sebagai metode pengendali. Kontrol Proporsional Integral digunakan untuk menentukan besarnya kecepatan dan arah putar motor DC sebagai penggerak, berdasarkan sudut kemiringan badan robot terhadap permukaan bidang datar. Sehingga ROBA-G1 dapat mempertahankan posisinya tegak lurus dengan seimbang terhadap permukaan bumi pada bidang datar.

1.2 Rumusan Masalah

Bagaimana membuat sebuah robot dapat berdiri tegak lurus menggunakan fusion sensor *accelerometer* dengan sensor *gyroscope* dengan metode *complementary filter*?

1.3 Batasan Masalah

- 1 Sensor yang digunakan adalah MPU6050 6 DOF Axis.
- 2 Menggunakan *Compelementary Filter* untuk menggabungkan pembacaan sensor *gyroscope* dan *accelerometer*.
- 3 Robot menggunakan motor DC disisi kanan dan kiri robot.
- 4 Robot mempertahankan posisi robot pada kondisi seimbang terhadap permukaan bumi yang datar.
- 5 Sistem kontrol yang digunakan adalah PD, dan tuning PD dengan *caratry and error*.
- 6 Pemograman menggunakan BascomAVR

1.4 Tujuan Penelitian

1. Membuat miniature/simulasi alat transportasi menggunakan dua roda yang dapat menyeimbangkan dirinya sendiri.
2. Mengimplementasikan fusion sensor *accelerometer* yang dapat menentukan kemiringan dan sensor *gyroscope* untuk kecepatan sudut.

1.5 Manfaat Penelitian

1. Memanfaatkan sensor *gyroscope* dan *accelerometer* sebagai input robot dalam menentukan arah putar dan kecepatan motor untuk mempertahankan posisi robot tetap seimbang.
2. Sebagai dasar teknologi untuk membuat alat transportasi yang dapat menyeimbangkan dirinya sendiri dengan dua roda.

1.6 Sistematika Penulisan

Untuk mempermudah pembahasan dan pemahaman maka sistematika penulisan dalam skripsi ini sebagai berikut:

BAB IV. PENCUNAN ALAT DAN PROGRAM

Pada bab ini meliputi hasil pengujian dari rangkaian dan cara kerja sistem serta penjelasan mengenai program yang digunakan.

BAB I. PENDAHULUAN

Dalam bab ini berisikan mengenai latar belakang, tujuan penelitian, batasan masalah, manfaat penelitian serta sistematika penulisan.

BAB II. TEORI DASAR DAN PENDUKUNG

Landasan teori, dalam bab ini dijelaskan tentang teori pendukung yang digunakan dalam pembahasan. Teori pendukung itu antara lain mikrokontroller ATMEGA16, Penggunaan dan setting pada software Bascom AVR.

BAB III. RANCANGAN SISTEM

Pada bab ini akan dibahas perancangan dari alat, yaitu diagram blok dari rangkaian, skematik dari masing-masing rangkaian dan diagram alir dari program yang akan diisikan ke mikrokontroller ATMega16

BAB IV. PENGUJIAN ALAT DAN PROGRAM

Pada bab ini meliputi hasil pengujian dari rangkaian dan cara kerja sistem serta penjelasan mengenai program yang digunakan.

BAB V. KESIMPULAN DAN SARAN

2.1 Penyeimbangan

Bab ini merupakan penutup yang meliputi tentang kesimpulan dari pembahasan yang dilakukan dari skripsi ini serta saran yang ditujukan demi perbaikan dan kesempurnaan kedepannya.

Salah satu dasar untuk membuat robot beroda dua dapat seimbang adalah dengan cara mengendalikan roda secara langsung dengan arah jatuhnya bagian atas sebuah robot. Apabila proses tersebut dapat terlaksana maka robot tersebut dapat seimbang.



Gambar 2.1. Pendulum terbalik di atas kerata beroda

Saat Penyeimbangan Robot Otomatis beroda dua condong ke depan pada gambar 2.1.1, maka tindakan yang perlu dilakukan

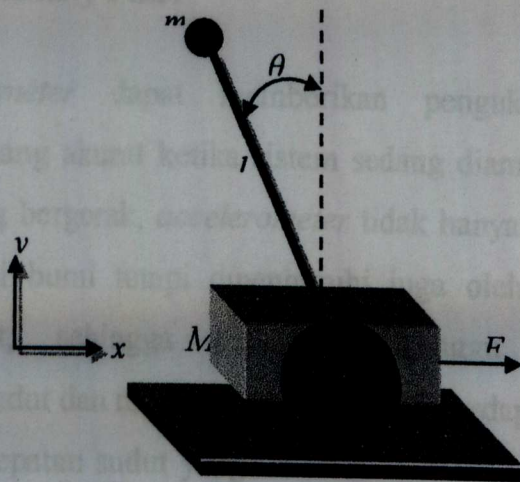
BAB II

TINJAUAN PUSTAKA

2.1 Penyeimbang Robot Otomatis

Penyeimbang Robot Otomatis merupakan pengembangan dari model pendulum terbalik yang diletakan di atas kereta beroda.

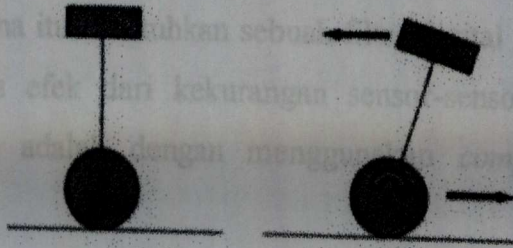
Ide dasar untuk membuat robot beroda dua dapat seimbang adalah dengan cara mengendalikan roda searah dengan arah jatuhnya bagian atas sebuah robot. Apabila proses tersebut dapat terlaksana maka robot tersebut dapat seimbang.



Gambar 2.1. Pendulum terbalik di atas kereta beroda

Saat Penyeimbang Robot Otomatis beroda dua condong ke depan pada gambar 2.1.1, maka tindakan yang perlu dilakukan

adalah motor akan memutar roda searah jarum jam.² Gaya yang digunakan untuk menyeimbangan dihasilkan dari putaran roda.



Gambar 2.2. Penyeimbang Robot Otomatis beroda dua menyeimbangkan diri

2.2 Complementary Filter

Accelerometer dapat memberikan pengukuran sudut kemiringan yang akurat ketika sistem sedang diam (statis). Bila sistem sedang bergerak, *accelerometer* tidak hanya dipengaruhi oleh gravitasi bumi tetapi dipengaruhi juga oleh pergerakan badan robot, sehingga dapat mengganggu pembacaan kemiringan sudut dan memiliki *noise*. *Gyroscope* dapat membaca data dari kecepatan sudut yang dinamis.³ Setelah integrasi data dari waktu ke waktu, perpindahan sudut atau sudut kemiringan

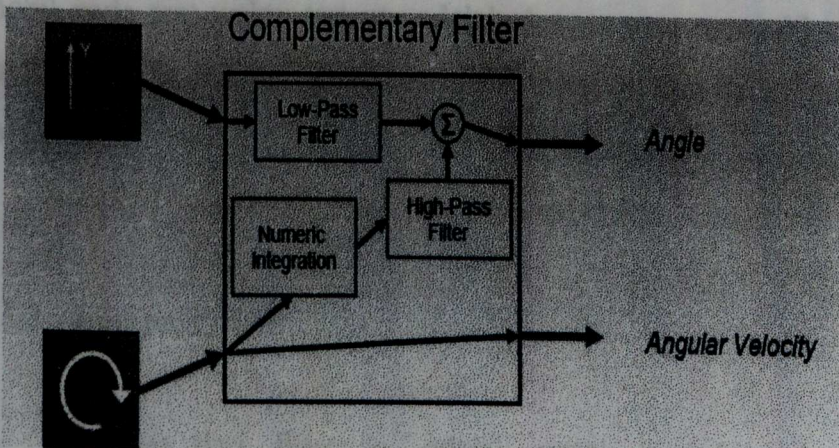
² Lio Prisko Ketaren, Mustaza Ma'a, Made Rahmawaty (2015), Balancing Robot Beroda Dua Menggunakan Metoda Kontrol Proporsional, Integral dan Derivatif

³Grasser, Felix; D'arrigo, Aldo; Colombi, Silvio; Rufer, Alfred (2002), "JOE: A Mobile, Inverted Pendulum", *IEEE Transactions on Industrial Electronics*, Vol. 49

dapat dihitung. Tetapi sudut ini akan menjadi tidak akurat dalam jangka panjang karena efek bias yang dihasilkan oleh *gyroscope*.

Oleh karena itu dibutuhkan sebuah filter digital yang dapat menghilangkan efek dari kekurangan sensor-sensor tersebut. Salah satunya adalah dengan menggunakan *complementary filter*.

Selain *complementary filter*, filter yang biasa digunakan adalah kalman filter, tetapi dalam hal ini hasil yang didapat signifikan maka dalam penulisan ini menggunakan *complementary filter* karena perhitungannya lebih singkat.



Gambar 2.3. Blok diagram *complementary filter*

Complementary filter adalah program fusion dari dua input dalam hal ini adalah output *accelerometer* dan output *gyroscope* untuk mendapatkan sebuah hasil penggabungan tersebut dalam

hal ini adalah sudut yang seharusnya, sehinggal menghasilkan sudut yang lebih akurat jika disbanding dengan pembacaan *accelerometer* saja atau dengan *gyroscope* saja.⁴

Konsep dasar pada complementary filter adalah *integration*, *low pass filter*, *high pas filter*, sampling periode dan waktu konstan. Adalah penjelasan konsep dasarnya sebagai berikut:

2.2.1 *Integration*

Konsep dasarnya sangat sederhana, kita analogikan seperti sebuah mobil yang bergerak dengan kecepatan yang telah diketahui, dan untuk mengetahui jarak yang ditempuh oleh mobil pada saat waktu tertentu adalah dengan cara mengalikan interval waktu dengan kecepatan mobil tersebut, dan dalam hal ini adalah :

$$\text{Angle} = \text{gyro} \times dt \dots \dots \dots (1)$$

2.2.2 *Low pass filter*

Tujuan dari filter ini adalah untuk menghilangkan fluktuasi kecil pada pada *accelerometer* dan melewati output dengan perubahan yang besar, dengan cara melakukan memaksakan perubahan sedikit demi sedikit dari sudut yang lalu

⁴Grasser Felix; D'arrigo Aldo, Colombi Silvio, Rufer Alfred (2002), "JOE: A Mobile, Inverted Pendulum", *IEEE Transactions on Industrial Electronics*, Vol. 49, No. 1

memaksakan perubahan sedikit demi sedikit dari sudut yang lalu secara terus menerus, dan jika dibuat secara matematis maka rumusnya adalah:

$$\text{angle} = (0.02) \times \text{acc} \dots \dots \dots (2)$$

2.2.3 High pass filter

Konsep ini lebih sulit dijelaskan jika dibandingkan dengan *low pass filter*, konsep dasar *high pass filter* bukan kebalikan dari *low pass filter*, konsep ini hanya memungkinkan sinyal dengan durasi pendek saja yang dilewatkan, hal ini berguna untuk menghilangkan drift saat robot mencapai titik seimbangnya dengan kecepatan yang tinggi.

2.2.4 Sampling periode

Sampling periode adalah interval waktu pengulangan program. Jika simple periode adalah 100 Hz maka interval waktu pengulangan program adalah 0,01 s atau 10 ms.

2.2.5 Waktu konstan

Waktu konstan adalah durasi relatif untuk mengeksekusi aksi, untuk *low pass filter* sinyal yang dilewatkan hanya sinyal yang lebih panjang dari waktu konstan, dan sebaliknya untuk *high pass filter*.

Lebih jauh tentang complementary filter, dalam waktu yang pendek pembacaan *gyroscope* lebih dominan dan meredam noise dari *accelerometer*, dan dalam waktu yang panjang yang lebih dominan adalah pembacaan *accelerometer* dan *gyroscope* berguna untuk meredam drift.

Adapun rumus dari *complementary filter* adalah:

$$\text{sudut} = (a) \times (\text{sudut} + \text{out_gyro} \times dt) + (1-a) \times (\text{sudut_acc})$$

.....(3)

Integration *Low pass filter* pada *accelerometer* *High pass filter*

Dimana:

a = koefisien filter

dt = waktu *sampling*, disesuaikan dengan waktu *sampling* nilai sensor

sudut = sudut keluaran *complementary filter*

out_gyro = keluaran sensor *gyroscope* berupa kecepatan sudut

sudut_acc = keluaran sensor *accelerometer* yang sudah berupa sudut

2.3 Pengendali PID

Sistem Kontrol PID(*Proportional-Integral-Derivative controller*) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan

robot adalah sebuah sistem yang membutuhkan presisi dan menekan noise sekecil mungkin dari bias pergerakan robot tersebut maka penggunaan kontrol PID sangat dibutuhkan agar aksi yang dilakukan lebih presisi dan lebih halus sehingga noise dari bias pergerakan robot dapat diminimalisir dan kecepatan dan arah yang dilakukan sesuai dengan error sudut robot.

Sistem kontrol PID terdiri dari tiga buah cara pengaturan yaitu kontrol P (Proportional), D (Derivative) dan I (Integral), dengan masing-masing memiliki kelebihan dan kekurangan.⁵ Dalam implementasinya masing-masing cara dapat bekerja sendiri maupun gabungan diantaranya. Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter P, I atau D agar tanggapan sinyal keluaran sistem terhadap masukan tertentu sebagaimana yang diinginkan

2.3.1 Kontrol Proporsional

Kontrol P jika $G(s) = k_p$, dengan k adalah konstanta. Jika $u = G(s) \times e$ maka $u = K_p \times e$ dengan K_p adalah Konstanta Proporsional. K_p berlaku sebagai Gain (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana

⁵ Bambang Nurcahyono, Sumardi, Budi Setiyono 2010, Self-Balancing Scooter Menggunakan Metode Kendali Proporsional Integral Derivatif

Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol P ini cukup mampu untuk memperbaiki respon transien khususnya rise time dan settling time.

2.3.2 Kontrol Integral

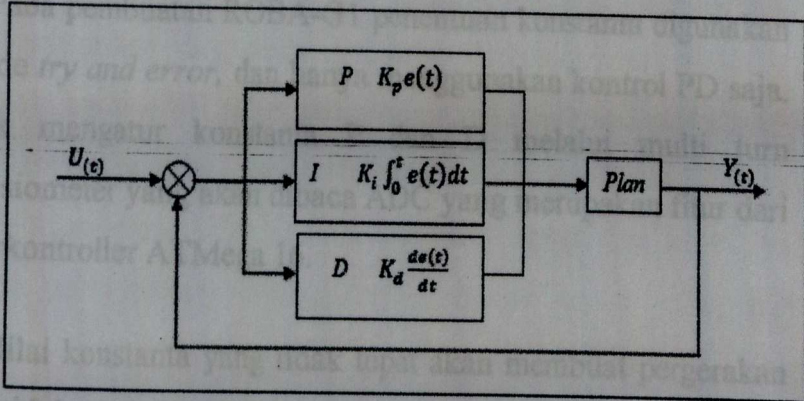
Jika $G(s)$ adalah kontrol I, maka u dapat dinyatakan sebagai $u(t) = [\text{integral}(t) \times dT]K_i$ dengan K_i adalah konstanta Integral, dan dari persamaan diatas, $G(s)$ dapat dinyatakan sebagai $u = K_i \cdot [\Delta t / \Delta t]$ Jika $e(T)$ mendekati konstan (bukan nol) maka $u(t)$ akan menjadi sangat besar sehingga diharapkan dapat memperbaiki error. Jika $e(T)$ mendekati nol maka efek kontrol I ini semakin kecil. Kontrol I dapat memperbaiki sekaligus menghilangkan respon steady-state, namun pemilihan K_i yang tidak tepat dapat menyebabkan respon transien yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan K_i yang sangat tinggi justru dapat menyebabkan output berosilasi karena menambah orde sistem, sehingga dalam pembuatan robot ini tidak menggunakan kontrol integral mengingat setting kontrol menggunakan metode *try and error*.

2.3.3 Kontrol Derivatif

Sinyal kontrol u yang dihasilkan oleh kontrol D dapat dinyatakan sebagai $G(s) = s \times K_d$ Dari persamaan di atas, nampak bahwa sifat dari kontrol D ini dalam konteks "kecepatan" atau rate dari error. Dengan sifat ini ia dapat digunakan untuk memperbaiki respon transien dengan memprediksi error yang akan terjadi. Kontrol Derivative hanya berubah saat ada perubahan error sehingga saat error statis

kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan kontroler Derivative tidak dapat dipakai sendiri⁶

Secara umum blok diagram dari kontrol PID adalah sebagai berikut:



Gambar 2.4. Block Diagram PID kontrol

Dalam waktu kontinyu, sinyal keluaran PD dapat dirumuskan sebagai berikut:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (4)$$

Dengan:

⁶ Bambang Nurcahyono, Sumardi, Budi Setiyono 2010, Self-Balancing Scooter Menggunakan Metode Kendali Proporsional Integral Derivatif

$u(t)$ = Sinyal keluaran pengendali PD

K_p = Konstanta Proporsional

K_i = Konstanta Integral

K_d = Konstanta Derivatif

$e(t)$ = Error

Pada pembuatan ROBA-G1 penentuan konstanta digunakan metode *try and error*, dan hanya menggunakan kontrol PD saja. Untuk mengatur konstanta P dan D melalui multi turn potensiometer yang akan dibaca ADC yang merupakan fitur dari mikorkontroller ATmega 16.

Nilai konstanta yang tidak tepat akan membuat pergerakan robot tidak setabil, maka perlu waktu untuk mengatur konstanta P dan D, tetapi karena input pengaturan konstanta menggunakan multi turn potensiometer maka dapat dilakukan walaupun robot sedang bekerja sehingga reaksi dapat terlihat dalam waktu yang sama. Dan itu dapat mempermudah pekerjaan untuk memilih konstanta P dan D dan dapat menghemat waktu.⁷

Nilai konstanta yang tepat akan pergerakan robot lebih halus, dan dapat menghilangkan gerakan yang kasar jika dibandingkan dengan tidak menggunakan sistem kontrol, jika hal ini terjadi maka tujuan penggunaan sistem kontrol ini untuk

⁷Suhata, ST, (2004), VB sebagai Pusat Kendali Peralatan Elektronik. Jakarta. PT.; Elex Media Komputindo.

memperhalus gerakan dan menentukan nilai output yang sesuai dengan error terpenuhi.

2.4 Mikrokontroller ATmega16

AVR merupakan salah satu jenis mikrokontroller yang di dalamnya terdapat berbagai macam fungsi. Perbedaannya dengan mikro yang pada umumnya digunakan seperti MCS 51 adalah pada AVR tidak perlu menggunakan oscillator eksternal karena di dalamnya sudah terdapat internal oscillator.⁸ Selain itu kelebihan dari AVR adalah memiliki Power-On Reset, yaitu tidak perlu adanya tombol reset dari luar karena cukup hanya dengan mematikan supply, maka secara otomatis AVR akan melakukan reset. Untuk beberapa jenis AVR terdapat beberapa fungsi khusus seperti ADC, EEPROM sekitar 512 bytes. Dalam hal ini yang digunakan adalah AVR ATMEGA16, perbedaannya dengan AVR ATMEGA16L hanyalah terletak pada besarnya tegangan yang diperlukan untuk bekerja. Untuk ATMEGA16 tipe L dapat bekerja pada tegangan antara 2,7V – 5,5V sedangkan untuk ATMEGA16 hanya dapat bekerja pada tegangan 4,5V – 5,5V.

Kelebihan dari ATMEGA16 sehingga digunakan sebagai kontrol utama adalah sebagai berikut;

⁸Mokh. Sholihul Hadi, (2003), Mengenal Mikrokontroler AVR ATmega16

Kelebihan dari ATMEGA16 sehingga digunakan sebagai kontrol utama adalah sebagai berikut;

- Mempunyai performa yang tinggi(berkecepatan akses maksimum 16MHz)dan hemat daya
- Memori untuk program flash cukup besaryaitu 16K Byte
- Memori internal SRAM sebesar 1K Byte
- EEPROM sebesar 512 byte yang dapatdiprogram saat operasi
- Port komunikasi SPI, Serial, I2C, One Wire dll
- Komunikasi serial standar USART
- Tersedia 4 chanel PWM
- Tersedia 3 chanel timer/counter (2 untuk 8bits dan 1 untuk 16 bits)

Berikut adalah gambar dari blok diagram untuk ATMEGA16:

Spesifikasi

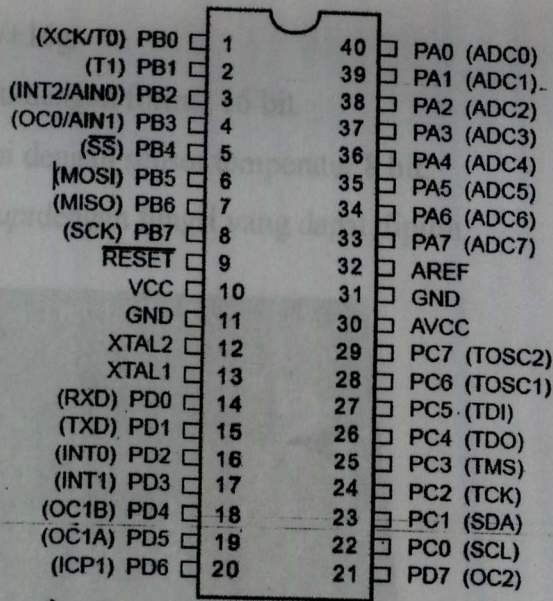
1. Tegangan kerja 3.3 VDC

2. Mampu membaca kecepatan sudut pada 6 sumbu.

3. Gyroscope range:

$\pm 250/\pm 500/\pm 1000/\pm 2000$ deg (degree per second).

4. Accelerometer range:



Gambar 2.5. Pin Konfigurasi

2.5 Sensor MPU 6050

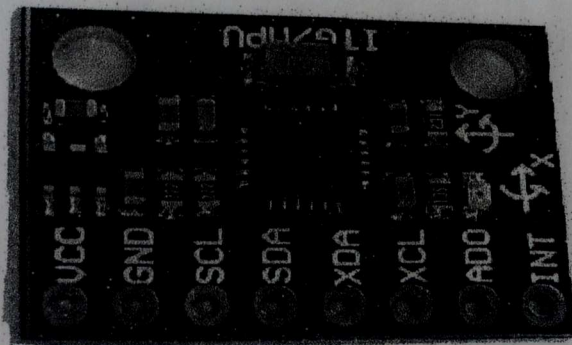
Robot ini menggunakan modul GY-521 yang menggunakan sensor MPU6050, sensor ini merupakan sensor 3 axis *gyroscope* dan 3 axis *accelerometer* dalam satu chip, pembacaan sensor ini cukup akurat, lebar bit yang digunakan adalah 16 bit untuk setiap channel. Dan untuk seri 6050 komunikasi hanya menggunakan I2C (sedangkan untuk seri yang lain dapat juga menggunakan komunikasi data SPI.

Spesifikasi:

1. Tegangan kerja 3,3 VDC.
2. Mampu membaca kecepatan sudut pada 6 sumbu.
3. *Gyroscope range*:
 $\pm 250/\pm 500/\pm 1000/\pm 2000$ dps (*degree per second*).
4. *Accelerometer range*:

output $\pm 2/\pm 4/\pm 8/\pm 16g$.

5. Data output dengan format 16 bit.
6. Dilengkapi dengan sensor temperatur 8 bit.
7. Pin *interrupt* dengan fungsi yang dapat dipilih.



Gambar 2.6. MPU6050

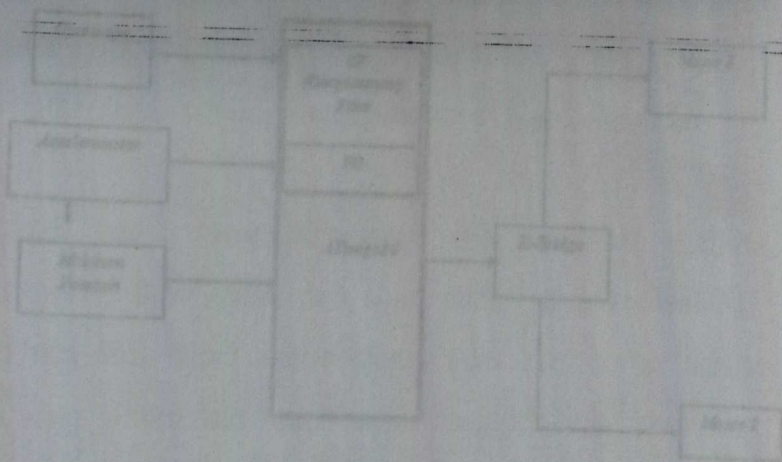
Tapi dalam pembuatan Roba_G1 hanya menggunakan 8 bit saja dan hanya menggunakan 2 (axis), sumbu Y untuk *accelerometer* dan sumbu Y untuk *Gyroscope*. Untuk menentukan sudut kemiringan hanya berdasarkan sumbu Y terhadap magnet bumi, tapi akan menjadi lebih baik jika sumbu Y berdasarkan sumbu X, tetapi untuk mengurangi down time pengoperasian program maka hanya menggunakan sumbu Y saja.

Komunikasi yang digunakan adalah komunikasi I2C, sehingga dapat menghemat port pada mikrokontroller, dan dapat mempermudah pendataan. Data yang diterima hanya berbentuk data digital dan tidak dapat langsung digunakan, perlu dikonversi agar dapat diproses ke *Complementary Filter*, untuk

output accelerometer dikonversi ke bentuk sudut sedangkan output gyroscope dikonversi ke bentuk radian.

3.1 Diagram Blok

Secara garis besar, diagram blok alat digambarkan pada gambar berikut ini:



Gambar 3.1 Diagram Blok Rangkaian

Pada diagram block di atas terdapat tiga input, input accelerometer, input gyroscope dan input setting PD dalam hal ini menggunakan multi turn potensiometer, untuk output motor didrive oleh H-Bridge.

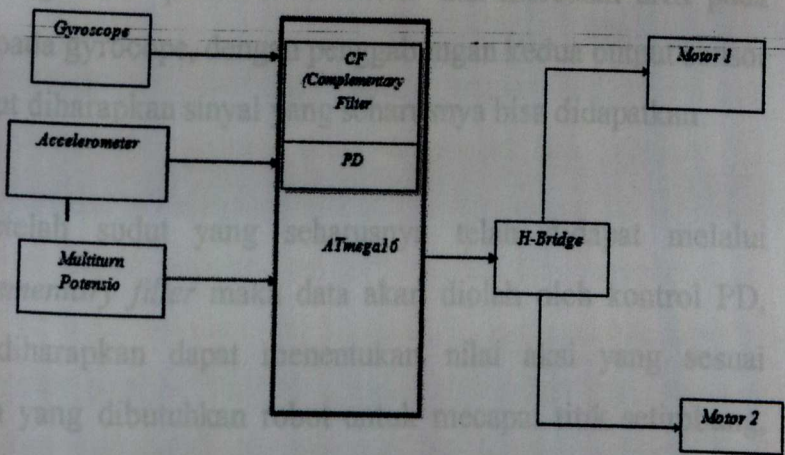
Pembacaan sensor accelerometer dan sensor gyroscope diterima oleh mikrokontroler menggunakan komunikasi data

BAB III

PERANCANGAN ALAT DAN PROGRAM

3.1 Diagram Blok

Secara garis besar, diagram blok alat digambarkan pada gambar berikut ini:



Gambar 3.1 Diagram Blok Rangkaian

Pada diagram block di atas terdapat tiga input, input accelerometer, input gyroscope dan input setting PD dalam hal ini menggunakan multi turn potensiometer, untuk output motor didrive oleh H-Bridge.

Pembacaan sensor *accelerometer* dan sensor *gyroscope* diterima oleh miktrokontoller menggunakan komunikasi data

I2C, yang hanya menggunakan dua port yaitu port SDA dan SCL.

Data *accelerometer* dan *gyroscope* yang diterima oleh mikrokontroller akan digabungkan dengan metode *compelementary filter*, untuk menghasilkan sudut yang seharusnya, *compelementary filter* menghasilkan output yang halus dengan cara meredam sinyal yang fluktuatif dan cenderung kecil pada *accelerometer* dan meredam drift pada robot pada *gyroscope*, dengan penggabungan kedua output sensor tersebut diharapkan sinyal yang seharusnya bisa didapatkan

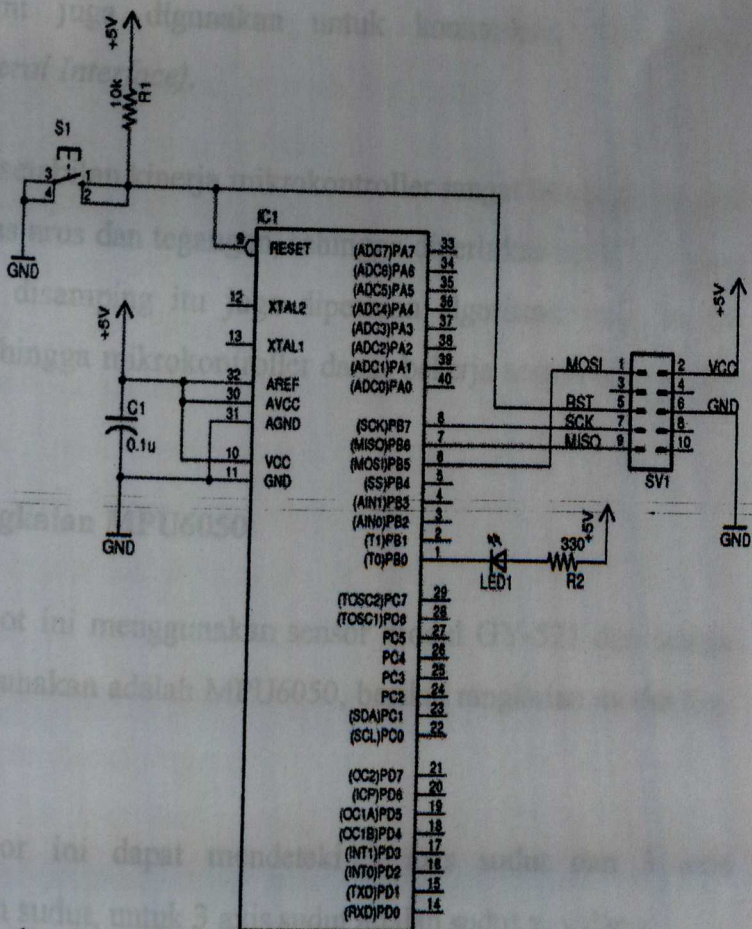
Setelah sudut yang seharusnya telah didapat melalui *complementary filter* maka data akan diolah oleh kontrol PD, yang diharapkan dapat menentukan nilai aksi yang sesuai dengan yang dibutuhkan robot untuk mencapai titik setimbang, dalam menentukan konstanta P dan D dapat diatur dengan memutar multiturn potensiometer, output dari multi turn potensiometer berupa tegangan analog sehingga dibutuhkan ADC yang terdapat dalam fitur mikrokontroler untuk membaca input dari mutli turn potensiometer, cara ini lebih efesien karena hasil output yang terlihat dalam bentuk gerakan motor akan terlihat saat itu juga, sehingga dapat mempermudah kita dalam menentuka konstanta P dan D yang tepat.

Agar ROBA-G1 dapat mencapai tujuannya untuk mempertahankan pada posisi seimbang maka diperlukan motor

yang didrive oleh H-Bridge, arah dan kecepatan motor dikontrol oleh sistem kontrol P dan D, feed back sistem kontrol tersebut adalah sensor *accelerometer* dan *gyroscope*. Sistem kontrol akan terus memperbaiki arah dan kecepatan motor sesuai dengan sudut robot, sehingga tujuan robot untuk mempertahankan diri dapat tercapai.

3.2 Perancangan Rangkaian Mikrokontroller ATmega16

Rangkaian ini berfungsi untuk mengendalikan seluruh sistem. Komponen utama dari rangkaian ini adalah IC mikrokontroler ATmega16. Pada IC ini program diisi sehingga rangkaian dapat berjalan sesuai dengan yang dikehendaki. Mikrokontroller ini memiliki 32 port I/O, yaitu port A, port B, port C dan port D. Pin 33 sampai 40 adalah Port A yang merupakan port ADC, dimana port ini dapat menerima data analog. Untuk supply tegangan Pin 10 dihubungkan ke sumber tegangan 5 volt sedangkan pin 11 dan pin 22 dihubungkan ke ground, dan Pin 32 dihubungkan ke tegangan 3,3V sebagai tegangan referensi ADC. Rangkaian mikrokontroller ditunjukkan pada gambar berikut ini:



Gambar 3.2 Rangkaian mikrokontroler ATmega16

Rangkaian mikrokontroler ini menggunakan komponen kristal sebagai sumber clocknya pada pin 12 dan pin 13. Nilai kristal ini akan mempengaruhi kecepatan mikrokontroler dalam mengeksekusi suatu perintah tertentu.

Untuk menulis flash rom pada mikrokontroler dapat menggunakan ISP Programmer seperti USBAsp, MKII, STK dan lain-lain, port yang digunakan untuk menulis flash rom adalah PortB5(MOSI), PortB6(MISO), PortB7(SCK), dan Reset,

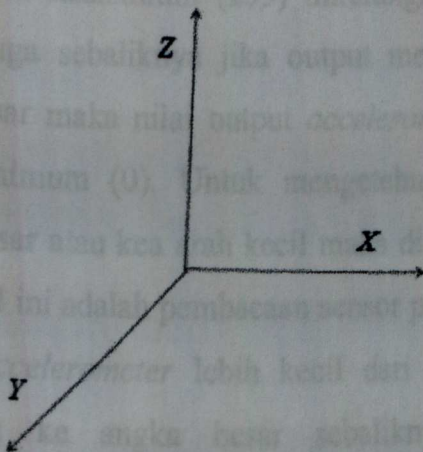
Port ini juga digunakan untuk komunikasi SPI (*Serial Peripheral Interface*).

Kesetabilan kinerja mikrokontroller sangat berpengaruh dari stabilitas arus dan tegangan, sehingga diperlukan regulator yang handal, disamping itu juga diperluan algoritma yang handal juga, sehingga mikrokontroller dapat bekerja secara efesian dan stabil.

3.3 Rangkaian MPU6050

Robot ini menggunakan sensor modul GY-521 dan sensor yang digunakan adalah MPU6050, berikut rangkaian modul Gy-521.

Sensor ini dapat mendeteksi 3 axis sudut dan 3 axis kecepatan sudut, untuk 3 axis sudut adalah sudut x, y dan z.



Gambar 3.3. Axis sudut Accelerometer

Pada robot ini hanya menggunakan satu axis sudut, yaitu sumbu y. data yang akan berubah jika sensor digerakkan searah sumbu y, data yang dihasilkan akan dibagi dengan sensitivitas sensor. Dalam perancangan ini hanya menggunakan 8 bit (0 – 255). Hasil pembacaan akan dibagi dengan sensitivitas untuk mendapat hasil dalam satuan g (gravitasi), dan nilai maksimumnya adalah 1 g dan minimumnya adalah -1 g, pada sensor ini nilai maksimal didapat pada sudut 90^0 jika digerakkan ke arah sudut 180 maka nilainya akan berkurang secara linier sesuai dengan sudut hingga ke sudut 0^0 sebaliknya jika digerakkan ke sudut 0^0 maka nilainya akan dimulai dari 0 dan naik secara linier hingga ke nilai 255 di sudut 0^0 , mengingat karakter pembacaan sensor untuk range 0 – 255 adalah $0^0 - 180^0$, dan nilai tertinggi dan terendah pada sudut 90 , maka perlu manipulasi program untuk mendapatkan sudut satu lingkaran penuh (00 – 360), jika output mengarah ke angka yang lebih kecil maka nilai maksimum (255) dikurangi output *accelerometer*, begitu juga sebaliknya jika output mengarah ke angka yang lebih besar maka nilai output *accelerometer* dikurangi dengan nilai minimum (0). Untuk mengetahui output mengarah ke angka besar atau ke arah kecil maka diperlukan angka referensi dalam hal ini adalah pembacaan sensor pada sudut 90^0 , jika nilai output *accelerometer* lebih kecil dari referensi maka nilainya mengarah ke angka besar sebaliknya jika nilai output *accelerometer* lebih besar dari nilai referensi maka nilainya

mengarah ke angka yang lebih kecil. Secara algoritma dapat ditulis sebagai berikut:

```
if Acc_y > then Acc_ref then
```

```
Acc_y = Acc_y - 0
```

```
Else
```

```
Acc_y = 0 - Acc_y
```

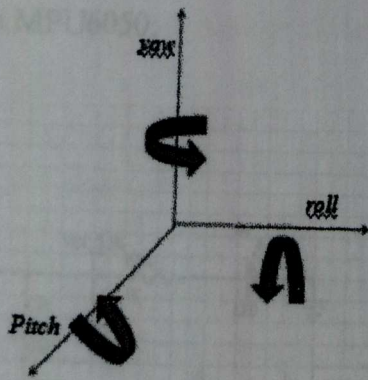
```
End if
```

Agar mendapat nilai satuan g, maka pembacaannya dibagi dengan 255, dan untuk vector arah maka dibuat offset sudut, dan sudut stabil adalah 90^0 . Untuk mengubah satuan g ke derajat maka harus diubah dulu ke bentuk radian (trigonometri pada mikrokontroller menghasilkan nilai pada satuan radian) dan untuk mengubah ke derajat maka nilai radian dilkali dengan $180/\pi$ secara matematis rumus yang digunakan untuk mendapatkan sudut adalah:

```
Acc_y = offset_y - Acc_y
```

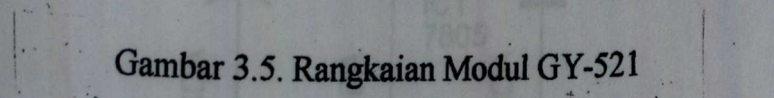
```
Angle = asin(Acc_y/255)*180/ $\pi$ 
```

Pada *gyroscope* 3 axis kecepatan sudut yang dibaca adalah Roll (x), Pitch (y) dan Yaw (z)



Gambar 3.4. Axis kecepatan sudut

Untuk kecepatan sudut yang digunakan adalah axis pitch (y), nilai yang dihasilkan sensor saat stabil adalah 255 atau 0, nilai hanya akan berubah jika ada pergerakan pada sumbu pitch, jika bergerak ke arah kanan maka nilai berkurang dari nilai stabil (255) sesuai dengan kecepatan pergerakan sensor yang bergerak di sumbu pitch dan jika bergerak di ke arah sebaliknya maka nilainya akan bertambah dari nilai nilai stabil (0). Dan nilai output yang dihasilkan setelah dikurangi oleh offset maka akan dibagi oleh sensitivitas sensor, untuk mendapatkan ke dalam satuan radian agar dapat dioleh oleh *Compelementary Filter*, Output *gyroscope* berpengaruh kepada kestabilan robot saat robot bergerak dan saat robot mencapai titik seimbang robot, itu karena output *gyroscope* akan berbalik saat terjadi robot berhenti mendadak dan besar kecilnya nilai juga berpengaruh dari seberapa cepat kecepatan sudutnya.



dihubungkan ke mikrokontroller, dan itu berguna untuk menghemat penggunaan port pada mikrokontroller, tetapi kekurangannya waktu yang dibutuhkan untuk membaca pastinya lebih lama jika dibandingkan dengan menggunakan portd ADC, tetapi down time tidak terlalu mempengaruhi keseimbangan robot, sehingga komunikasi menggunakan ini dapat digunakan.

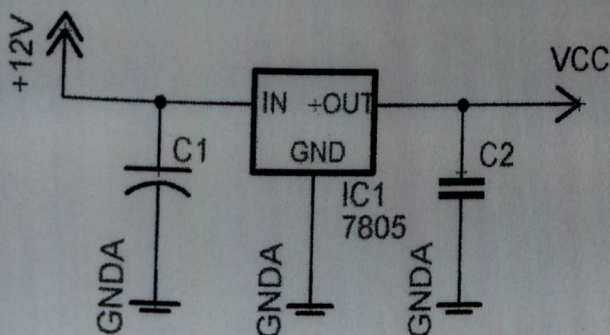
menghasilkan tegangan 3,3 Volt di dalam modul terlah

ditambahkan regulator 3,3Volt sehingga kita tidak perlu lagi menambahkan rangkaian regulator tambahan.

3.4 Rangkaian Regulator

Agar rangkaian dapat bekerja, maka dibutuhkan sumber tegangan. Pada perancangan ini dibutuhkan sumber tegangan yang konstan, yaitu (+) 5 volt DC, yang digunakan untuk menyuply tegangan ke IC ATmega16 dan sensor MPU6050.

Rangkaian tampak seperti gambar di bawah ini:



Gambar 3.6. Rangkaian Regulator

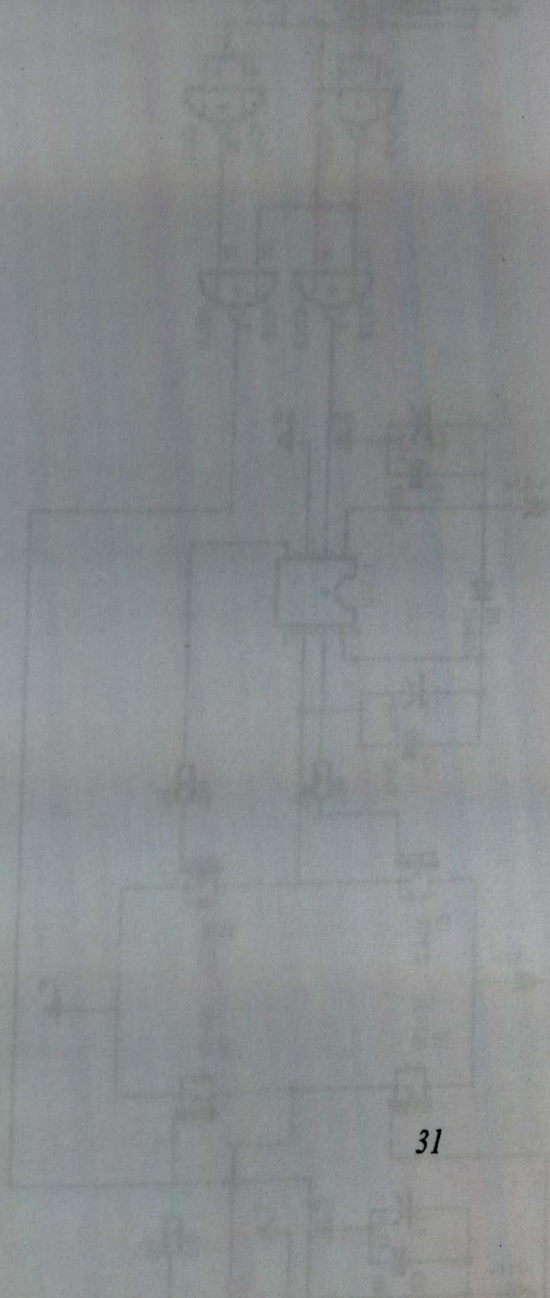
Rangkaian di atas cukup stabil, dan komponen yang digunakan juga minim, sehingga PCB yang digunakan dapat diminimalisir.

3.5 Rangkaian H-Bridge

Rangkaian H-bridge digunakan untuk mengatur arah putaran motor dan kecepatan motor, arah putaran diatur dengan cara mengendalikan transistor-transistor yang disusun

membentuk huruf H, untuk driver mosfet digunakan IR2111,
berikut rangkaian H-Bridge yang digunakan pada ROBA1

Gambar 3.7 Rangkaian H-Bridge



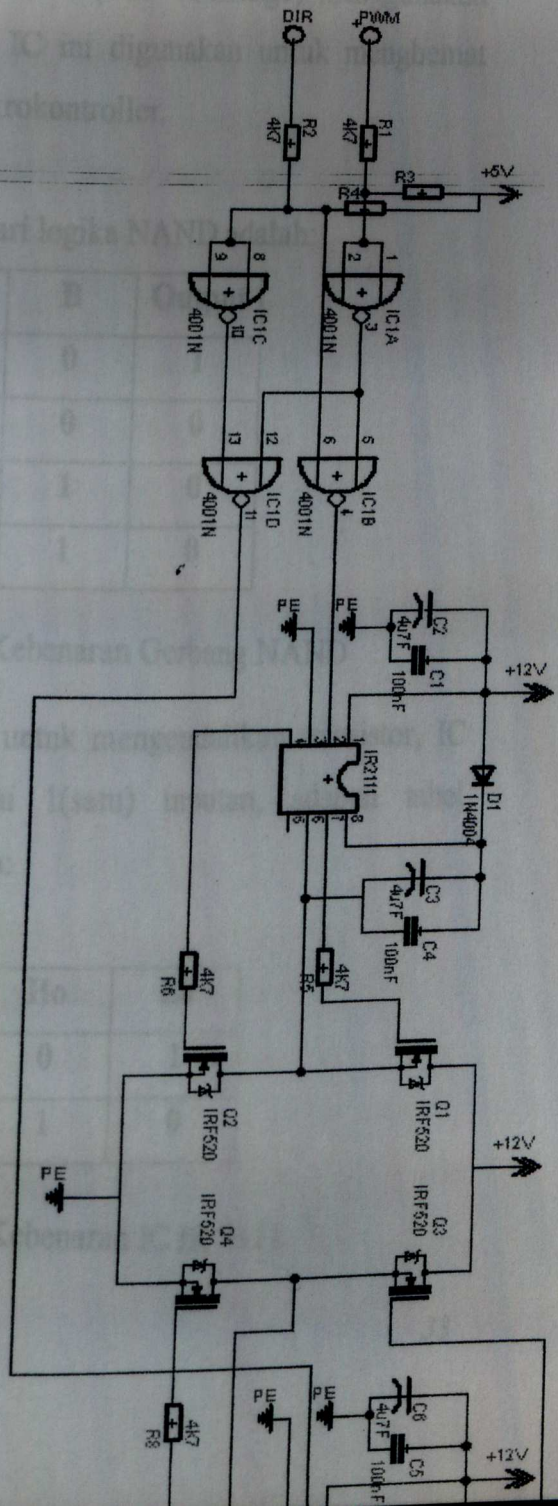
Dari rangkaian di atas hanya 1(satu) rangkaian H-Bridge, sedangkan penggunaan ROBO-G1 menggunakan 2 (dua) rangkaian H-Bridge.

Untuk mengontrol IC2111 (Half H-Bridge) menggunakan IC logika NAND 4001, IC ini digunakan untuk menghantarkan port pada mikrokontroler.

Adapun tabel kebenaran dari logika NAND

A	B	C
0	0	0
1	0	0
0	1	1
1	1	1

Gambar 3.7. Rangkaian H-Bridge



Dari rangkaian di atas hanya 1(satu) rangkaian H-Bridge, sedangkan penggunaan ROBO-G1 menggunakan 2 (dua) rangkaian H-Bridge.

Untuk mengontrol IC2111 (Half H-Bridge) menggunakan IC logika NAND 4001, IC ini digunakan untuk menghemat penggunaan port pada mikrokontroller.

Adapun tabel kebenaran dari logika NAND adalah:

A	B	Output
0	0	1
1	0	0
0	1	0
1	1	0

Tabel 3.1. Tabel Kebenaran Gerbang NAND

IC IR2111 digunakan untuk mengendalikan transistor, IC IR2111 hanya mempunyai 1(satu) inputan, adapun tabel kebenaran IC IR2111 adalah:

Input	Ho	Lo
0	0	1
1	1	0

Tabel 3.2. Tabel Kebenaran IC IR 2111

Untuk mengendalikan PWM IC yang digunakan IC 4001 A dan IC 4001 B, sedangkan untuk mengendalikan arah putaran motor digunakan IC 4001 C dan IC 4001 D, kedua input IC 4001 digabungkan dan kedua input IC 4001 C digabungkan.

Jika logika 0 diberikan pada input PWM, maka output IC 4001 bernilai 1, output dari IC 4001 A yang berlogika 1 menjadi inputan IC 4001 B dan satu inputan lagi dari inputan arah yang berlogika 0, input logika 0 dan 1 pada IC 4001 B akan menghasilkan output 0 pada IC 4001 B, dan jika input arah berlogika 0 maka output IC 4001 C adalah berlogika 1, output IC 4001 C akan menjadi salah satu inputan pada IC 4001 D, dan inputan satu lagi dari inputan PWM yang berlogika 0, input logika 1 dan 0 pada IC 4001 D akan menghasilkan output 0 pada IC 4001 D.

Output yang berlogika 0 dari IC 4001 B akan menjadi inputan pada IC IR2111 A, sehingga output Ho akan bernilai 0 dan output Lo berlogika 1 pada IC IR2111 A akan bernilai 1, hal membuat transistor Q1 akan cut off dan Q2 akan saturasi. Output yang berlogika 0 dari IC 4001 D akan menjadi inputan pada IC IR2111 B, sehingga output Ho akan bernilai 0 dan output Lo berlogika 1 pada IC IR2111 B akan bernilai 1, hal membuat transistor Q3 akan cut off dan Q4 akan saturasi. Sehingga arus tidak dapat mengalir dari Q1 dan Q3 dan motor tidak berputar.

Jika logika 1 diberikan pada input PWM, maka output IC 4001 bernilai 0, output dari IC 4001 A yang berlogika 0 menjadi inputan IC 4001 B dan satu inputan lagi dari inputan arah yang berlogika 0, input logika 0 dan 0 pada IC 4001 B akan menghasilkan output 1 pada IC 4001 B, dan jika input arah berlogika 0 maka output IC 4001 C adalah berlogika 1, output IC 4001 C akan menjadi salah 1 inputan pada IC 4001 D, dan inputan satu lagi dari inputan PWM yang berlogika 0, input logika 1 dan 0 pada IC 4001 D akan menghasilkan output 0 pada IC 4001 D.

Output yang berlogika 1 dari IC 4001 B akan menjadi inputan pada IC IR2111 A, sehingga output Ho akan bernilai 1 dan output Lo akan berlogika 0 pada IC IR2111 A, hal membuat transistor Q1 akan saturasi dan Q2 akan cut off. Output yang berlogika 0 dari IC 4001 D akan menjadi inputan pada IC IR2111 B, sehingga output Ho akan bernilai 0 dan output Lo berlogika 1 pada IC IR2111 B akan bernilai 1, hal membuat transistor Q3 akan cut off dan Q4 akan saturasi. Sehingga arus mengalir dari Q1 ke Q4 maka motor akan berputar searah CW.

Jika logika 0 diberikan pada input PWM, maka output IC 4001 bernilai 1, output dari IC 4001 A yang berlogika 1 menjadi inputan IC 4001 B dan satu inputan lagi dari inputan arah yang

berlogika 0, input logika 0 dan 1 pada IC 4001 B akan menghasilkan output 0 pada IC 4001 B, dan jika input arah berlogi 1 maka output IC 4001 C adalah berlogika 0, output IC 4001 C akan menjadi salah satu inputan pada IC 4001 D, dan inputan satu lagi dari inputan PWM yang berlogi 0, input logika 0 dan 0 pada IC 4001 D akan menghasilkan output 1 pada IC 4001 D.

Output yang berlogika 0 dari IC 4001 B akan menjadi inputan pada IC IR2111 A, sehingga output Ho akan bernilai 0 dan output Lo berlogika 1 pada IC IR2111 A, hal membuat transistor Q1 akan cut off dan Q2 akan saturasi. Output yang berlogika 1 dari IC 4001 D akan menjadi inputan pada IC IR2111 B, sehingga output Ho akan bernilai 1 dan output Lo berlogika 0 pada IC IR2111 B akan bernilai 1, hal membuat transistor Q3 akan saturasi dan Q4 akan cut off. Sehingga arus mengalir dari Q3 ke Q2 dan motor berputar searah CCW.

Jika logika 1 diberikan pada input PWM, maka output IC 4001 bernilai 0, output dari IC 4001 A yang berlogika 0 menjadi inputan IC 4001 B dan satu inputan lagi dari inputan arah yang berlogika 1, input logika 1 dan 1 pada IC 4001 B akan menghasilkan output 0 pada IC 4001 B, dan jika input arah berlogika 1 maka output IC 4001 C adalah berlogika 0, output IC 4001 C akan menjadi salah satu inputan pada IC 4001 D, dan inputan satu lagi dari inputan PWM yang berlogi 1, input logika

1 dan 1 pada IC 4001 D akan menghasilkan output 0 pada IC 4001 D.

Output yang berlogika 1 dari IC 4001 B akan menjadi inputan pada IC IR2111 A, sehingga output Ho akan bernilai 1 dan output Lo akan berlogika 0 pada IC IR2111 A akan bernilai 1, hal membuat transistor Q1 akan saturasi dan Q2 akan cut off. Output yang berlogika 1 dari IC 4001 D akan menjadi inputan pada IC IR2111 B, sehingga output Ho akan bernilai 1 dan output Lo berlogika 0 pada IC IR2111 B akan bernilai 1, hal membuat transistor Q3 akan saturasi dan Q4 akan cut off. Sehingga arus mengalir dari Q1 ke Q4 dan arus mengalir dari Q3 ke Q2 maka motor tidak akan berputar.

Jadi kesimpulan logika pengaturan kecepatan dan arah motor dapat dibuat dalam sebuah tabel kebenaran sebagai berikut:

Control		Motor	
Direction	PWM	Motor	Direction
0	0	OFF	-

0	1	ON	CW
1	0	OFF	-
1	1	ON	CCW

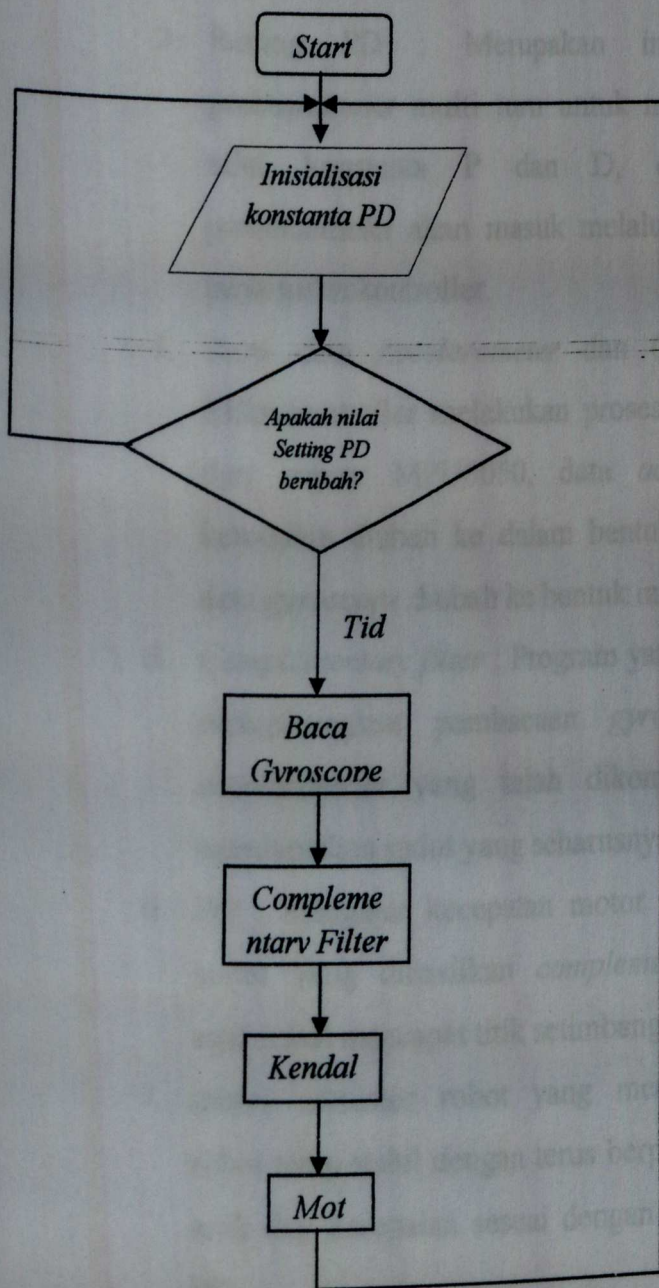
Tabel 3.3. Tabel kebenaran rangkaian H-Bridge

Dengan demikian dengan bantuan IC logika NAND dapat mempermudah kita dalam memprogram karena hanya menggunakan 1 (satu) output untuk mengendalikan arah (direction) dan satu lagi port yang berfungsi sebagai PWM.

Transistor yang digunakan adalah jenis mosfet IRF 520, penggunaan mosfet pada perancangan H-Bridge untuk mempercepat proses switching, dan arus yang mengalir lebih besar, tetapi karena menggunakan motor dengan arus yang tidak besar maka penggunaan mosfet difokuskan kepada kecepatan switching.

Pada mosfet IFR 520 telah dilengkapi oleh diode didalamnya, sehingga pada perancangan rangkaian H-Bridge tidak perlu lagi menambahkan diode.

3.6 Perancangan Program



Gambar 3.8. Flowchart program

Keterangan :

1. Start : Memulai Program.
2. Inisialisasi konstanta P dan D
3. Setting PD : Merupakan inputan dari potensiometer multi turn untuk mengendalikan nilai konstanta P dan D, output dari potensiometer akan masuk melalui port ADC pada mikrokontroller.
4. Baca data *Accelerometer* dan *Gyroscope* : Mikrokontroller melakukan proses pembacaan dari sensor MPU6050, data *accelerometer* kemudian diubah ke dalam bentuk sudut dan data *gyroscope* diubah ke bentuk radian.
5. *Complementary filter* : Program yang berfungsi menggabungkan pembacaan *gyroscope* dan *accelerometer* yang telah dikonversi untuk mendapatkan sudut yang seharusnya.
6. PD : mengatur kecepatan motor sesuai hasil sudut yang dihasilkan *complementary filter* agar robot mencapai titik setimbang.
7. motor : actuator robot yang memungkinkan robot tetap stabil dengan terus berputar dengan arah dan kecepatan sesuai dengan output data PD.
8. Program terus loop ke setting PD.

BAB IV

PENGUJIAN ALAT DAN PROGRAM

Pada bab ini, akan dibahas pengujian alat mulai dari pengujian alat permodul sampai pengujian alat secara keseluruhan. Pengujian tersebut akan dilakukan secara bertahap dengan urutan sebagai berikut:

- a. Pengujian masing-masing blok
- b. Pengujian sistem secara keseluruhan

4.1 Pengujian Mikrokontroller ATMEGA16

Untuk mengetahui apakah rangkaian mikrokontroller bekerja dengan baik, maka dilakukan pengujian. Pengujian pada bagian ini dilakukan dengan memberikan program pada mikrokontroller. Programnya adalah sebagai berikut :

=====Regfile mikrokontroller=====

\$regfile = "m16def.dat"

\$crystal = 11059200

=====Konfigurasi Pin=====

DDRC = 255

=====Main Program=====

Do

PORTC = 255

Waitms 1000

PORTC = 0

Waitms 1000

loop

Jika program tersebut dijalankan, maka Port C akan bernilai "high" selama satu detik dan "low" selama satu detik, dengan demikian maka system minimum mikrokontroller ATMEGA16 telah berjalan dengan baik.

4.2 Pengujian Gyroscope

Untuk mengetahui apakah modul GY-521 3 axis gyroscope bekerja dengan baik, maka dilakukan pengujian. Pengujian pada bagian ini dilakukan dengan memberikan program pada mikrontroller. Programnya adalah sebagai berikut :

```
'=====Regfile mikrokontroller=====
```

```
$regfile = "m16def.dat"
```

```
$crystal = 11059200
```

```
$baud = 9600
```

```
'=====Konfigurasi Variabel=====
```

```
dim Y as byte
```

```
dim Y1 as single
```

```
dim angle_rate as single
```

```
'===== Register MPU6050=====
```

```
Const MPU_R = &HD0
```

```
Const MPU_W = &HD1
```

```
Config Scl = Portc.0
```

```
Config Sda = Portc.1
```

```
I2cinit
```

```

I2cstart                                ' Generate start code
I2cwbyte MPU_w
I2cwbyte &H19
I2cwbyte &H07                            ' send address
I2cstop
Waitms 2

```

```

I2cstart                                ' Generate start code
I2cwbyte Adxl_w
I2cwbyte &H1B
I2cwbyte &H00                            ' send address
I2cstop
Waitms 2

```

```

I2cstart                                ' Generate start code
I2cwbyte MPU_w
I2cwbyte &H6B
I2cwbyte &H00                            ' send address
I2cstop
Waitms 2

```

```

=====Main Program=====

```

```

Do
I2cstart
    I2cwbyte MPU_W

```



```

I2cwrite &H45
I2cstart
I2cwrite MPU_R
I2cwrite Y, Nack
I2cstop
If Y < 100 Then
Y1 = 0 - Y
Else
Y1 = 255 - Y1
End If
Angle_rate = Y1 / 255
Angle_rate = Angle_rate * 100
Print "Data ADC : ";Y1," ","Velocity : ";Angle_rate
Waitms 500
Return

```

Data dikirim menggunakan komunikasi serial ke computer (PC) untuk melihat hasil pembacaan sensor.

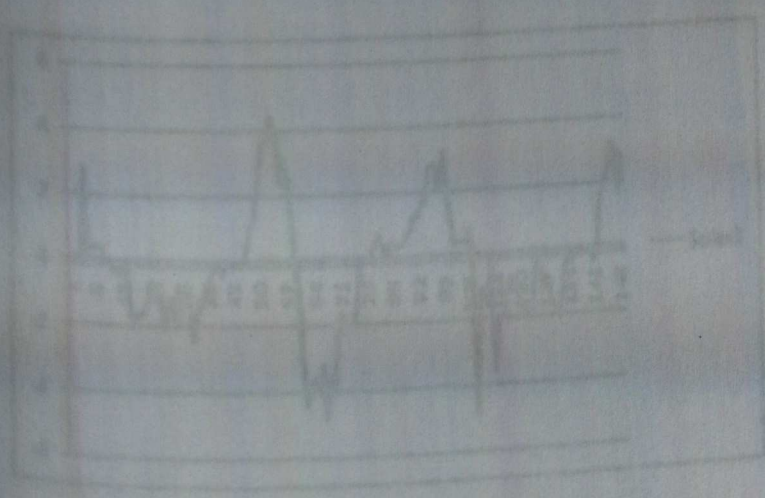
Ketika sensor digerakkan ke depan dan ke belakang, maka pembacaan sensornya secara grafik adalah pada gambar 4.2, sementara data yang didapat dari pembacaan sensor terlihat pada table 4.2.1. Data yang di dapat telah dikonversi ke bentuk rad/s.

Besar kecilnya pembacaan sensor bergantung pada arah

Tabel 4.1 Hasil pembacaan gyroscope

gerakan, dan kecepatan gerakan. Pada gambar 4.2.1 terlihat nilai velocity yang bernilai positif dan bernilai negatif, dan value juga berbeda, itu terjadi karena sensor digerakkan ke arah depan sehingga hasil pembacaan sensor akan bernilai positif, karena nilai pembacaan lebih besar dari nilai offset sensor, begitu sebaliknya jika sensor digerakkan ke arah belakang maka hasilnya pembacaannya velocity akan bernilai negatif, besar kecilnya value bergantung kepada cepat atau lambatnya kita menggerakkan sensor, nilai offset didapat saat pembacaan sensor pada kondisi diam.

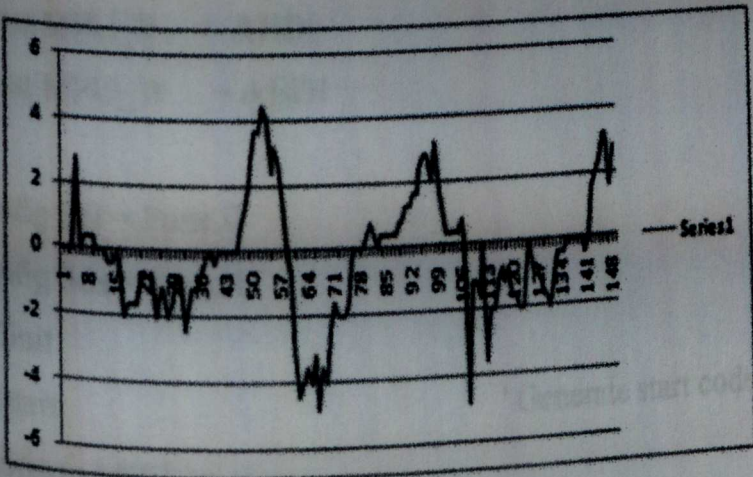
1	0.0	0.0	0.0	0.0	0.0
2	0.1	0.2	0.3	0.4	0.5
3	0.2	0.3	0.4	0.5	0.6
4	0.3	0.4	0.5	0.6	0.7
5	0.4	0.5	0.6	0.7	0.8
6	0.5	0.6	0.7	0.8	0.9
7	0.6	0.7	0.8	0.9	1.0
8	0.7	0.8	0.9	1.0	1.1
9	0.8	0.9	1.0	1.1	1.2
10	0.9	1.0	1.1	1.2	1.3
11	1.0	1.1	1.2	1.3	1.4
12	1.1	1.2	1.3	1.4	1.5
13	1.2	1.3	1.4	1.5	1.6
14	1.3	1.4	1.5	1.6	1.7
15	1.4	1.5	1.6	1.7	1.8
16	1.5	1.6	1.7	1.8	1.9
17	1.6	1.7	1.8	1.9	2.0
18	1.7	1.8	1.9	2.0	2.1
19	1.8	1.9	2.0	2.1	2.2
20	1.9	2.0	2.1	2.2	2.3
21	2.0	2.1	2.2	2.3	2.4
22	2.1	2.2	2.3	2.4	2.5
23	2.2	2.3	2.4	2.5	2.6
24	2.3	2.4	2.5	2.6	2.7
25	2.4	2.5	2.6	2.7	2.8
26	2.5	2.6	2.7	2.8	2.9
27	2.6	2.7	2.8	2.9	3.0
28	2.7	2.8	2.9	3.0	3.1
29	2.8	2.9	3.0	3.1	3.2
30	2.9	3.0	3.1	3.2	3.3
31	3.0	3.1	3.2	3.3	3.4
32	3.1	3.2	3.3	3.4	3.5
33	3.2	3.3	3.4	3.5	3.6
34	3.3	3.4	3.5	3.6	3.7
35	3.4	3.5	3.6	3.7	3.8
36	3.5	3.6	3.7	3.8	3.9
37	3.6	3.7	3.8	3.9	4.0
38	3.7	3.8	3.9	4.0	4.1
39	3.8	3.9	4.0	4.1	4.2
40	3.9	4.0	4.1	4.2	4.3
41	4.0	4.1	4.2	4.3	4.4
42	4.1	4.2	4.3	4.4	4.5
43	4.2	4.3	4.4	4.5	4.6
44	4.3	4.4	4.5	4.6	4.7
45	4.4	4.5	4.6	4.7	4.8
46	4.5	4.6	4.7	4.8	4.9
47	4.6	4.7	4.8	4.9	5.0
48	4.7	4.8	4.9	5.0	5.1
49	4.8	4.9	5.0	5.1	5.2
50	4.9	5.0	5.1	5.2	5.3
51	5.0	5.1	5.2	5.3	5.4
52	5.1	5.2	5.3	5.4	5.5
53	5.2	5.3	5.4	5.5	5.6
54	5.3	5.4	5.5	5.6	5.7
55	5.4	5.5	5.6	5.7	5.8
56	5.5	5.6	5.7	5.8	5.9
57	5.6	5.7	5.8	5.9	6.0
58	5.7	5.8	5.9	6.0	6.1
59	5.8	5.9	6.0	6.1	6.2
60	5.9	6.0	6.1	6.2	6.3
61	6.0	6.1	6.2	6.3	6.4
62	6.1	6.2	6.3	6.4	6.5
63	6.2	6.3	6.4	6.5	6.6
64	6.3	6.4	6.5	6.6	6.7
65	6.4	6.5	6.6	6.7	6.8
66	6.5	6.6	6.7	6.8	6.9
67	6.6	6.7	6.8	6.9	7.0
68	6.7	6.8	6.9	7.0	7.1
69	6.8	6.9	7.0	7.1	7.2
70	6.9	7.0	7.1	7.2	7.3
71	7.0	7.1	7.2	7.3	7.4
72	7.1	7.2	7.3	7.4	7.5
73	7.2	7.3	7.4	7.5	7.6
74	7.3	7.4	7.5	7.6	7.7
75	7.4	7.5	7.6	7.7	7.8
76	7.5	7.6	7.7	7.8	7.9
77	7.6	7.7	7.8	7.9	8.0
78	7.7	7.8	7.9	8.0	8.1
79	7.8	7.9	8.0	8.1	8.2
80	7.9	8.0	8.1	8.2	8.3
81	8.0	8.1	8.2	8.3	8.4
82	8.1	8.2	8.3	8.4	8.5
83	8.2	8.3	8.4	8.5	8.6
84	8.3	8.4	8.5	8.6	8.7
85	8.4	8.5	8.6	8.7	8.8
86	8.5	8.6	8.7	8.8	8.9
87	8.6	8.7	8.8	8.9	9.0
88	8.7	8.8	8.9	9.0	9.1
89	8.8	8.9	9.0	9.1	9.2
90	8.9	9.0	9.1	9.2	9.3
91	9.0	9.1	9.2	9.3	9.4
92	9.1	9.2	9.3	9.4	9.5
93	9.2	9.3	9.4	9.5	9.6
94	9.3	9.4	9.5	9.6	9.7
95	9.4	9.5	9.6	9.7	9.8
96	9.5	9.6	9.7	9.8	9.9
97	9.6	9.7	9.8	9.9	10.0
98	9.7	9.8	9.9	10.0	10.1
99	9.8	9.9	10.0	10.1	10.2
100	9.9	10.0	10.1	10.2	10.3



Gambar 4.1. Hasil pembacaan gyroscope secara grafik

Tabel 4.1 Hasil pembacaan *gyroscope*

No	Data	No	Data	No	Data	No	Data	No	Data
1	0	31	-0.8	61	-2.8	91	1.2	121	-1.6
2	0	32	-2.4	62	-4.4	92	1.2	122	-2
3	0	33	-2	63	-4	93	1.6	123	-2
4	0	34	-1.2	64	-3.6	94	1.6	124	-1.2
5	2.8	35	-1.2	65	-4	95	2.4	125	0
6	0	36	-0.8	66	-3.2	96	2.8	126	-0.8
7	0.4	37	-0.4	67	-4.8	97	2.8	127	-0.8
8	0.4	38	0	68	-3.6	98	2	128	-1.2
9	0.4	39	0	69	-4	99	3.2	129	-1.6
10	0	40	-0.4	70	-2.4	100	1.6	130	-1.6
11	0	41	0	71	-1.6	101	1.2	131	-2
12	0	42	0	72	-2	102	0.4	132	-1.2
13	-0.4	43	0	73	-2	103	0.4	133	-0.4
14	-0.4	44	0	74	-2	104	0.4	134	-0.4
15	0	45	0	75	-1.6	105	0.4	135	-0.4
16	-0.4	46	0	76	0	106	0.8	136	0
17	-2	47	0.8	77	0	107	0	137	0
18	-1.6	48	1.2	78	0	108	-4.8	138	0
19	-1.6	49	2	79	0	109	-1.2	139	0
20	-1.6	50	2.8	80	0.4	110	-1.2	140	0
21	-1.2	51	3.6	81	0.8	111	-1.6	141	-0.4
22	-0.8	52	3.6	82	0.4	112	0	142	0
23	-0.8	53	4.4	83	0	113	-3.6	143	1.6
24	-0.8	54	4	84	0.4	114	-2	144	2
25	-2	55	2.4	85	0.4	115	-2	145	2.4
26	-1.2	56	3.2	86	0.4	116	-1.2	146	3.2
27	-1.2	57	2.8	87	0.4	117	-0.8	147	3.2
28	-2	58	1.6	88	0.4	118	-1.2	148	1.6
29	-1.6	59	0	89	0.8	119	-0.8	149	2.8
30	-0.8	60	-0.8	90	0.8	120	-1.6	150	0



Gambar 4.1. Hasil pembacaan *gyroscope* secara grafik

4.3 Pengujian Accelerometer

Untuk mengetahui apakah rangkaian mikrokontroller bekerja dengan baik, maka dilakukan pengujian. Pengujian pada bagian ini dilakukan dengan memberikan program pada mikrokontroller. Programnya adalah sebagai berikut :

=====Regfile mikrokontroller=====

\$regfile = "m16def.dat"

\$crystal = 11059200

\$baud = 9600

=====Konfigurasi Variabel=====

dim Y as byte

dim Y1 as integer

dim angle as single

===== Register MPU6050=====

Const MPU_R = &HD0

Const MPU_W = &HD1

Config Scl = Portc.0

Config Sda = Portc.1

I2Cinit

I2Cstart

I2Cwrite MPU_w

I2Cwrite &H19

I2Cwrite &H07

' Generate start code

' send address

I2cstop

Waitms 2

I2cstart

' Generate start code

I2cwbyte Adxl_w

I2cwbyte &H1B

I2cwbyte &H00

' send address

I2cstop

Waitms 2

I2cstart

' Generate start code

I2cwbyte MPU_w

I2cwbyte &H6B

I2cwbyte &H00

' send address

I2cstop

Waitms 2

=====Main Program=====

Do

I2cstart

I2cwbyte MPU_W

I2cwbyte &H3D

I2cstart

I2cwbyte MPU_R

I2crbyte Y, Nack

I2cstop

If $Y < 100$ Then

$$Y1 = 0 - Y$$

Else

$$Y1 = 255 - Y$$

End If

=====Conver 2 Angle=====

$$Y1 = Y1 / 255$$

$$\text{Angle} = \text{Asin}(Y1)$$

$$\text{Angle} = 57.295 * \text{Angle}$$

Print "Data ADC : ";Y;" ";"Drajat : ";angle

Hasil pembacaanya menggunakan komunikasi serial ke komputer, karena digit desimal yang terlalu banyak, sehingga penggunaan LCD 2 X 16 tidak dapat digunakan dengan maksimal.

Disini juga terlihat hasil pembacaan sensor yang telah konversi data hasil pembacaan sensor ke bentuk drajat.

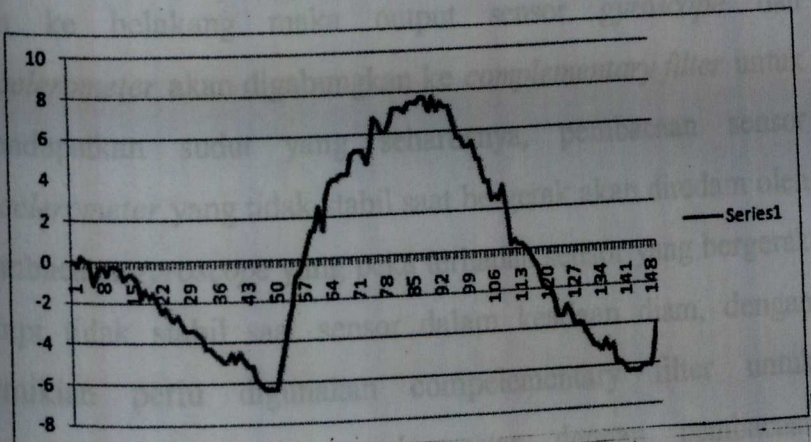
Sama seperti sensor *gyroscope*, pada tabel 4.3.1, nilai positif dan negatif pada pembacaan derajat berpengaruh dari arah sensor digerakan, jika sensor digerakkan ke depan maka hasil pembacaannya akan positif karena nilai pembacaan sensor lebih besa dari nilai offset dan sebaliknya jika sensor digerakkan ke belakang maka hasil pemabacaannya akan negatif karena

pembacaan sensor lebih kecil dari offset, nilai offset didapat dari pembacaan sensor pada sudut 90^0 .

Besar kecilnya nilai derajat bergantung dari sudut sensor dari sudut offset (90^0). Pada gambar 4.3.1 adalah hasil pembacaan sensor ketika sensor digeser ke arah depan dan ke belakang hingga sudut tertentu.

Tabel 4.2. Hasil pembacaan *accelerometer*

No	Data	No	Data	No	Data	No	Data	No	Data
1	0.00	31	-3.60	61	2.47	91	7.66	121	-2.02
2	-0.22	32	-3.82	62	1.35	92	6.98	122	-2.47
3	0.22	33	-4.27	63	2.92	93	7.44	123	-3.37
4	0.00	34	-4.27	64	3.60	94	7.21	124	-3.37
5	0.00	35	-4.50	65	3.82	95	7.21	125	-2.92
6	-1.12	36	-4.72	66	3.82	96	6.53	126	-3.15
7	-0.45	37	-4.95	67	4.05	97	5.63	127	-3.82
8	-0.22	38	-4.95	68	4.05	98	5.63	128	-3.82
9	-0.67	39	-4.50	69	4.05	99	5.17	129	-3.82
10	-0.22	40	-4.95	70	4.95	100	5.17	130	-4.27
11	-0.22	41	-4.50	71	5.17	101	5.40	131	-4.50
12	-0.45	42	-4.95	72	5.17	102	4.50	132	-4.95
13	-0.90	43	-5.17	73	5.17	103	4.50	133	-4.95
14	-0.67	44	-5.17	74	4.50	104	3.37	134	-5.17
15	-0.67	45	-5.85	75	5.40	105	3.37	135	-4.72
16	-0.90	46	-6.08	76	6.76	106	2.47	136	-5.17
17	-1.80	47	-6.30	77	6.53	107	2.70	137	-4.95
18	-1.35	48	-6.30	78	6.08	108	2.47	138	-5.85
19	-1.80	49	-6.30	79	6.08	109	3.15	139	-5.85
20	-1.57	50	-6.30	80	6.53	110	1.80	140	-6.08
21	-2.02	51	-5.17	81	7.21	111	0.22	141	-6.30
22	-2.47	52	-4.05	82	7.21	112	0.45	142	-6.30
23	-2.25	53	-3.15	83	7.21	113	0.22	143	-6.30
24	-2.70	54	-2.25	84	6.98	114	0.22	144	-6.30
25	-2.92	55	-0.90	85	7.44	115	0.00	145	-6.08
26	-2.70	56	-0.67	86	7.44	116	-0.22	146	-6.08
27	-3.15	57	0.00	87	7.44	117	-0.67	147	-6.08
28	-3.37	58	0.67	88	7.66	118	-0.90	148	-5.63
29	-3.37	59	1.35	89	7.66	119	-1.80	149	-4.05
30	-3.37	60	1.35	90	6.98	120	-2.02	150	0



Gambar 4.2. Hasil pembacaan *accelerometer* secara grafik

4.4 Pengujian *Complementary Filter*

Filter ini digunakan untuk mengabungkan pembacaan *gyroscope* dan *accelerometer*, potongan program *compelementery filter* berdasarkan rumus

$$\text{sudut} = (a) * (\text{sudut} + \text{out_gyro} * dt) + (1-a) * (\text{sudut_acc});$$

dimana $dt = 0,010$

$$a = 0,98$$

$$\text{degree} = 0$$

$$\text{Degree2} = \text{Rz1} * 0.010$$

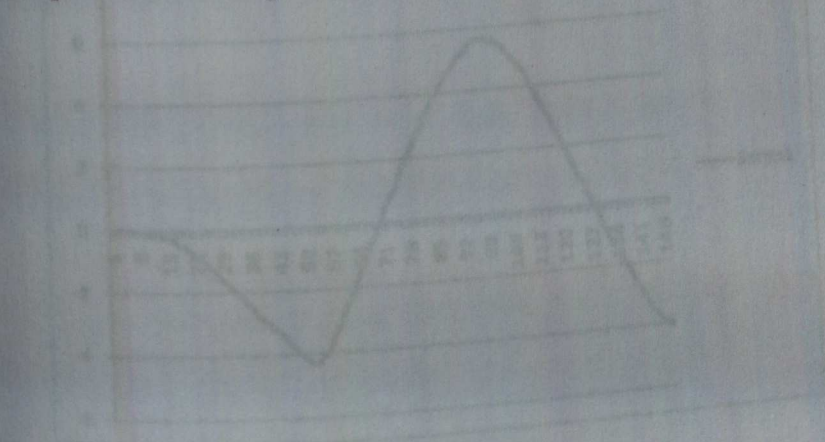
$$\text{Degree2} = \text{Degree} + \text{Degree2}$$

$$\text{Degree2} = \text{Degree} * 0.94$$

$$\text{Degree} = 0.06 * \text{Rx1}$$

$$\text{Degree} = \text{Degree} + \text{Degree2}$$

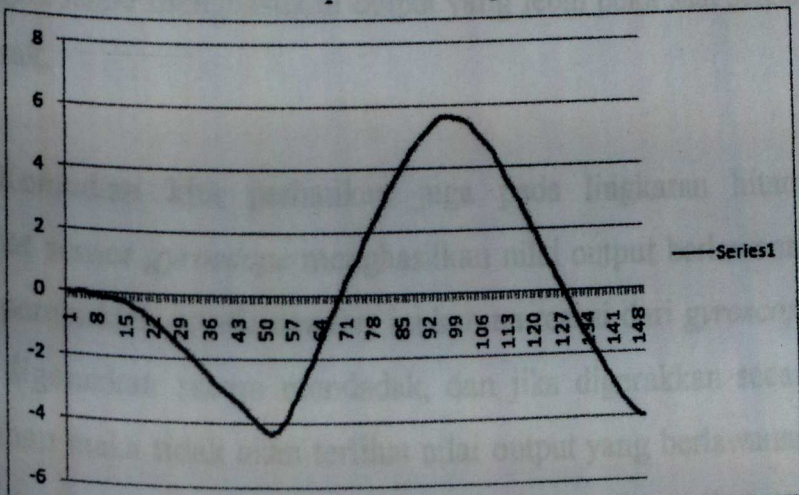
Ketika program dijalankan dan sensor digerakkan ke depan dan ke belakang maka output sensor *gyroscope* dan *accelerometer* akan digabungkan ke *complementary filter* untuk mendapatkan sudut yang seharusnya, pembacaan sensor *accelerometer* yang tidak stabil saat bergerak akan diredam oleh pembacaan *gyroscope* yang peka terhadap sensor yang bergerak tetapi tidak stabil saat sensor dalam keadaan diam, dengan demikian perlu digunakan *complementary filter* untuk menutupi kekurangan *accelerometer* dengan pembacaan *gyroscope*. Intinya *gyroscope* hanya mengoreksi sudut saat sensor digerakkan, sehingga aksi yang dilakukan berupa putaran motor akan berbanding proposional dengan sudut yang seharusnya, mengingat robot ini bersifat mobile dan mencapai keseimbangan maka hanya memperbolehkan sedikit toleransi saja, dalam kata lain kestabilan sensor dan algoritma fusion sensor dengan *complementary filter* sangat dibutuhkan. berikut hasil pembacaan *complementary filter* sebagai berikut:



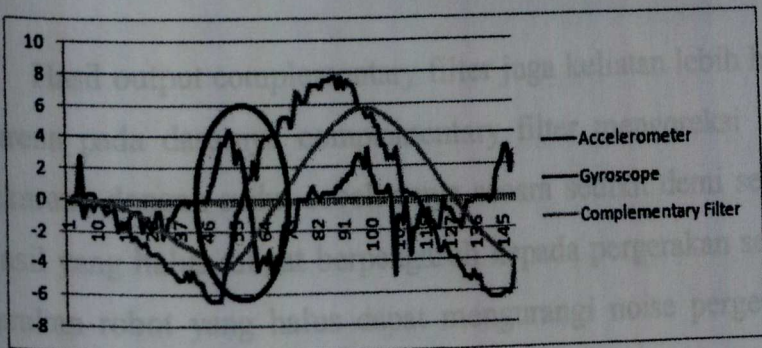
Gambar 4.3. Hasil pembacaan *Complementary Filter* dalam bentuk grafik

Tabel 4.3. Hasil pembacaan Compelementry Filter

No	Data	No	Data	No	Data	No	Data	No	Data
1	0.00	31	-1.84	61	-2.35	91	5.05	121	1.84
2	-0.02	32	-1.97	62	-2.15	92	5.17	122	1.57
3	0.00	33	-2.11	63	-1.87	93	5.31	123	1.26
4	0.00	34	-2.25	64	-1.56	94	5.43	124	0.98
5	0.01	35	-2.39	65	-1.25	95	5.55	125	0.74
6	-0.06	36	-2.53	66	-0.96	96	5.62	126	0.51
7	-0.08	37	-2.68	67	-0.68	97	5.64	127	0.24
8	-0.09	38	-2.82	68	-0.42	98	5.65	128	-0.01
9	-0.12	39	-2.92	69	-0.17	99	5.63	129	-0.24
10	-0.12	40	-3.04	70	0.13	100	5.61	130	-0.49
11	-0.13	41	-3.13	71	0.42	101	5.61	131	-0.74
12	-0.15	42	-3.24	72	0.70	102	5.54	132	-1.00
13	-0.20	43	-3.35	73	0.96	103	5.48	133	-1.24
14	-0.23	44	-3.46	74	1.16	104	5.36	134	-1.48
15	-0.25	45	-3.61	75	1.41	105	5.24	135	-1.67
16	-0.29	46	-3.76	76	1.73	106	5.08	136	-1.88
17	-0.39	47	-3.90	77	2.02	107	4.93	137	-2.07
18	-0.46	48	-4.04	78	2.26	108	4.76	138	-2.29
19	-0.55	49	-4.17	79	2.49	109	4.66	139	-2.51
20	-0.62	50	-4.28	80	2.73	110	4.48	140	-2.72
21	-0.71	51	-4.32	81	3.01	111	4.22	141	-2.94
22	-0.82	52	-4.29	82	3.26	112	3.99	142	-3.14
23	-0.91	53	-4.20	83	3.50	113	3.75	143	-3.32
24	-1.02	54	-4.06	84	3.71	114	3.53	144	-3.49
25	-1.14	55	-3.86	85	3.93	115	3.31	145	-3.64
26	-1.24	56	-3.65	86	4.15	116	3.09	146	-3.77
27	-1.36	57	-3.42	87	4.34	117	2.86	147	-3.89
28	-1.49	58	-3.17	88	4.55	118	2.63	148	-3.99
29	-1.61	59	-2.90	89	4.74	119	2.36	149	-3.98
30	-1.72	60	-2.65	90	4.88	120	2.09	150	0



Gambar 4.3. Hasil pembacaan Compelementry Filter dalam bentuk grafik



Gambar 4.4. Hasil pembacaan *Accelerometer*, *Gyroscope* dan *Complementary Filter* dalam bentuk grafik

Jika kita perhatikan dari grafik terlihat bahwa hasil *complementary filter* lebih cenderung mengikuti pergerakan pembacaan *accelerometer*, dan *gyroscope* lebih kepada meredam hasil pembacaan *accelerometer*. Mengingat akan terjadi bias pada pembacaan *accelerometer* saat sensor bergerak dan *gyroscope* menghasilkan output yang lebih peka saat sensor bergerak.

Kemudian kita perhatikan juga pada lingkaran hitam, terlihat sensor *gyroscope* menghasilkan nilai output berlawanan dari pembacaan *accelerometer*, ini karena reaksi dari *gyroscope* jika digerakkan secara mendadak, dan jika digerakkan secara perlahan maka tidak akan terlihat nilai output yang berlawanan, hal ini sangat berguna untuk meredam pergerakan robot saat robot mencapai titik setimbang dengan kecepatan motor yang tinggi.

Hasil output complementary filter juga kelihatan lebih halus, karena pada dasarnya complementary filter mengoreksi sudut sekarang dengan sudut sebelumnya secara sedikit demi sedikit. Hasil yang halus sangat berpengaruh kepada pergerakan sensor, gerakan robot yang halus dapat mengurangi noise pergerakan robot itu sendiri sehingga robot akan lebih stabil.

4.5 Pengujian PID

PID yang digunakan adalah program yang disadur dari rumus (2), pada aplikasi robot ini hanya menggunakan control P dan D saja. adapun potongan programnya:

Error = 0 - Degree

Delta_error = Error - Last_error

Derivatif = $K_d * \text{Delta_error}$

Sum_error = Last_error + Error

//Integral = $K_i * \text{Sum_error}$

//Integral = Integral / 2

Pd = Proporsional + Derivatif

//PD = Pd + Integral

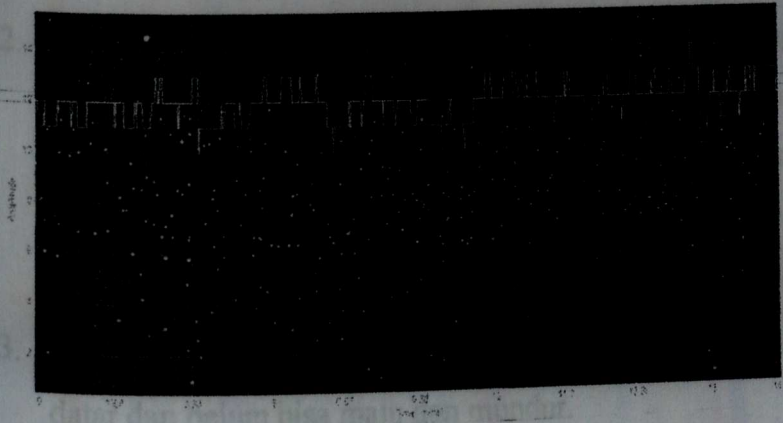
Last_error = Error

Kecepatan = Int(PD)

Pwm1a = kecepatan

Pwm1b = kecepatan

Kesetabilan robot **sangat** bergantung dari keakuratan hasil pembacaan **complementary filter** yang menjadi input PD dan penentuan konstanta P dan D. Dari gambar 4.6 terlihat **nois**, ini terjadi karena arus balik dari motor yang berputar berlawanan arah secara terus menerus, sehingga keliatan robot akan terus bergerak ke kiri dan ke kanan.



Gambar 4.5. PWM saat robot dalam keadaan stabil berdasarkan hasil perhitungan PD

1. Sebaiknya menggunakan motor yang mempunyai torsi dan kecepatan yang lebih cepat, agar robot dapat mempertahankan posisi di saat robot dalam keadaan miring, dan hal itu juga memungkinkan robot dapat berjalan di permukaan yang miring.
2. Roda dengan daya gesek tinggi sangat membantu robot untuk menahan dalam posisi seimbang, roda yang licin membuat robot sulit mencapai sudut seimbang.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Masih terjadi noise pada pwm motor, hal ini terjadi karena arus balik dari motor.
2. Robot hanya dapat mempertahankan posisi seimbang maksimal pada sudut -10^0 - 10^0 , hal ini terjadi karena torsi dan kecepatan motor tidak cukup cepat menyeimbangkan robot, dan tidak cukup kuat menahan beban robot.
3. Robot hanya dapat seimbang di atas permukaan yang datar dan belum bisa maju dan mundur.

5.2 Saran

1. Sebaiknya menggunakan motor yang mempunyai torsi dan kecepatan yang lebih cepat, agar robot dapat mempertahankan posisi disaat robot dalam keadaan miring, dan hal ini juga memungkinkan robot dapat berjalan di permukaan yang miring.
2. Roda dengan gaya gesek tinggi sangat membantu robot untuk menahan dalam posisi seimbang, roda yang licin membuat robot sulit mencapai sudut seimbang.

3. Pada Project selanjutnya akan dibuat dalam ukuran yang lebih besar hingga dapat digunakan sebagai alat transportasi yang ramah lingkungan dan hemat tempat.
4. Perlu algoritma yang lebih baik untuk menghasilkan pergerakan robot yang lebih halus dan robot tetap menjaga pada keadaan setimbang.

DAFTAR PUSTAKA

- Bambang Nurcahyono, Sumardi, Budi Setiyono 2010, Self-Balancings Scooter Menggunakan Metode Kendali Proporsional Integral Derivatif
- Grasser Felix; D'arrigo Aldo, Colombi Silvio, Rufer Alfred (2002), "JOE: A Mobile, Inverted Pendulum", *IEEE Transactions on Industrial Electronics*, Vol. 49, No. 1, pp. 107 - 114, 2002.
- Ketaren Prisko Lio, Ma'a Mustaza, Rahmawaty Made (2015), Balancing Robot Beroda Dua Menggunakan Metoda Kontrol Proporsional, Integral dan Derivatif
- Mokh. Sholihul Hadi, (2003), Mengenal Mikrokontroler AVR ATMega16
- Ogata, Katsuhiko, Jakarta. 1994. "Teknik Kontrol Automatik Jilid 1", Diterjemahkan Oleh Ir.Edi Leksono, Erlangga
- Setiawan, Iwan, Jakarta,2008."Kontrol PID untuk Proses Industri", Elex Media Komputindo,.
- Suhata, ST, (2004), VB sebagai Pusat Kendali Peralatan Elektronik. Jakarta. PT.; Elex Media Komputindo.